

COMPLETE LISTING OF THE CLAIMS**OFFICIAL**

1. (Previously Presented): A system for interacting with computer programming languages comprising:

a plurality of objects associated with a code model that encapsulate semantic elements of a plurality of programming languages and provide a language neutral interface thereto that insulates a programmer from unique syntaxes associated with a plurality of programming languages.

2. (Cancelled)

3. (Cancelled)

4. (Cancelled)

5. (Previously Presented) The system of claim 1, wherein the code model models at least one of a syntax, one or more functions, one or more objects, and one or more data structures associated with the computer programming language.

6. (Previously Presented) The system of claim 1, wherein the plurality of code model objects are defined by a set of object classes to enable modeling and abstraction of the computer programming language.

7. (Cancelled)

8. (Previously Presented) The system of claim 1, wherein the semantic elements include at least one of an attribute, a method, a type, an interface, a statement, an expression and a parameter.

9. (Original) The system of claim 6, wherein the set of object classes further comprise at least one of a project item, a project, a code model, a file code model, a code

09/686,033

MS154755.1

element, a code name space, a code type, a code class a code interface, a code enumeration, a code delegate, a code function, a code parameter, a code attribute, a code structure, a code variable, a code property, a code type reference.

10. (Original) The system of claim 9 further comprising a programmatic interface, wherein the programmatic interface includes methods for interacting with the set of object classes.

11. (Previously Presented) The system of claim 9, wherein code model objects associated with the computer programming languages are modified *via* at least one of an add code method, a delete code method, and coding *via* a text buffer.

12. (Original) The system of claim 9, further comprising a CodeVariable object for modeling variables within a project.

13. (Original) The system of claim 12, wherein the code variable object further comprises attributes that include at least one of an attribute for storing a parent of the CodeVariable object, an attribute for storing the initial expression and value for the variable modeled by the CodeVariable object, an attribute for storing the type of the variable modeled by the CodeVariable object, an attribute for storing whether the variable belongs to a class or to instances of a class, and an attribute for determining whether the variable is a constant.

14-21 (Cancelled)

22. (Previously Presented) A system for interacting with computer programming languages at a semantic and syntactic level comprising:

a code model associated with a plurality of code model objects to enable a programmatic interface to interact with a plurality of computer programming languages at a semantic and syntactic level in a language neutral manner, wherein the code model provides isolation between the programmatic interface and computer programming languages.

09/686,033

MS154755.1

23. (Original) The system of claim 22, wherein the code model objects encapsulate functionality within the computer programming languages.

24. (Previously Presented) The system of claim 23, wherein the code model describes at least one of a syntax, one or more functions, one or more objects, and one or more data structures associated with the computer programming languages.

25. (Original) The system of claim 23, wherein the code model objects are defined by a set of object classes to enable modeling and abstraction of the computer programming languages.

26. (Original) The system of claim 25, wherein the code model objects model semantic elements within the computer programming languages.

27. (Original) The system of claim 26, wherein the semantic elements include at least one of an attribute, a method, a type, an interface, and a parameter.

28. (Original) The system of claim 25, wherein the set of object classes further comprise at least one of a project item, a project, a code model, a file code model, a code element, a code name space, a code type, a code class, a code interface, a code enumeration, a code delegate, a code function, a code parameter, a code attribute, a code statement, a code expression, a code structure, a code variable, a code property, a code type reference.

29. (Original) The system of claim 25, wherein set of object classes associated with the computer programming languages are modified to become the code model objects *via* at least one of an add code method, a delete code method, and coding *via* a text buffer.

30. (Original) The system of claim 25, further comprising a CodeVariable object for modeling variables within a project.

09/686,033

MS154755.1

31. (Original) The system of claim 30, wherein the code variable object further comprises attributes that include at least one of an attribute for storing a parent of the CodeVariable object, an attribute for storing the initial expression and value for the variable modeled by the CodeVariable object, an attribute for storing the type of the variable modeled by the CodeVariable object, an attribute for storing whether the variable belongs to a class or to instances of a class, and an attribute for determining whether the variable is a constant.

32. (Previously Presented) A method for interacting with a computer programming language comprising:

- retrieving a code model object from a code model;
- interacting with the code model object to specify a semantic element;
- incorporating the code model object into a program project, the code model mapping the code model object to the correct syntax for a particular computer programming language associated with the semantic element.

33. (Previously Presented) The method of claim 32, the semantic element comprising one of a class, an object, a function, a method, an attribute, a variable, a delegate, a statement and a parameter.

34. (Previously Presented) The method of claim 32, further comprising modifying the code model mapping.

35. (Previously Presented) The method of claim 34, wherein the code model mapping is modified utilizing a text editor.